

# Contour Error Compensation based on Feed Rate Adjustment

Jiangang Li, Yiming Wang, Qin Shu, Deqing Huang and Yanan Li

**Abstract**—To improve the performance of computer numerical control (CNC) machining, especially for large-curvature trajectories, this paper presents a contour error compensation algorithm based on reference trajectory modification. In order to estimate the contour error accurately and efficiently, a contour error estimation model is established. The reference trajectory is modified on the basis of the estimated contour error and partitioned into different segments, which adopt different feed rates according to a corner detection algorithm. The effectiveness of this contour error compensation algorithm is verified by experiments on a CNC machine tool.

## I. INTRODUCTION

In computer numerical control (CNC) machining, due to delay characteristics of servo systems [1], [2] and dynamic mismatch between axes [3], [4], [5], machining accuracy is usually determined by the contour error (the shortest distance between the actual position and the reference contour). How to achieve the required contouring performance, especially in high-speed and large-curvature contouring tasks, has always been an important problem in manufacturing applications [6]. In general, contour control can be divided into two parts, i.e. contour error estimation and contour error control.

For linear and circular curves, the analytical solution of contour error estimation is relatively easier to compute, but it is difficult to apply to free-form curves. To address this issue, many researchers approximate them using linear or circular line segments, e.g. tangent line and osculating circle approximation methods respectively in [7] and [8], [9]. Affected by the tracking error, the tangent line approximation is a first-order approximation, so its performance deteriorates during high-speed tracking of contours with large curvatures. Since the circle approximation is a second-order approximation, the calculation of the osculating circle is time-consuming, leading to low efficiency. To overcome these limitations, the natural local approximation based contour error estimation methods are developed to improve the contouring accuracy of biaxial servo systems [10], [11].

In contour error control, cross-coupled control (CCC) is proposed to reduce the contour error in [12], which is then further modified in [13]. Since then, many improved control strategies based on CCC have been presented, such as biaxial

synchronization control in [14] and model-free adaptive contour control in [15]. In [16], CCC is implemented in multi-axis machine tools. Other than CCC, there are other contour error control methods, such as the discrete-time fractional-order sliding-mode contour error control method in [17]. Nevertheless, all of these methods require access to low-level controllers that is not allowed in many CNC systems.

For repetitive CNC machining tasks typical for products manufactured in batches, individual axis iterative learning control (ILC) and cross-coupled control are combined to improve motion control performance [18]. Applying the techniques of ILC to CCC enables learning of the cross-coupled error, which leads to a modified control signal and subsequent improvement in the contour tracking performance. While the ILC used in most of the existing works is based on a time period, the contour error is naturally defined in space. Therefore, any delay in one axis will introduce an error to the contour error compensation. In the field of ILC, spatial iterative learning control (sILC) is introduced for systems with spatial periodicity. In [19] and [20], the standard temporal ILC algorithm is formulated with spatial variables and sILC is developed based on two-dimensional spatial convolution. In [21], sILC is developed for a class of high-order nonlinear motion control systems in the presence of both parametric and nonparametric uncertainties, but it is also implemented in the control level. [11] develops sILC with modification of reference trajectory by modifying G code. While this method allows system improvement without interference of the low-level controllers, the performance on a large-curvature contour is relatively poor.

As [22] shows that the contour error is related to the feed rate and the curvature of the trajectory, the segments with large curvatures in a trajectory have large contour errors. In order to solve this problem, [23] proposes a feed rate adjustment strategy based on fuzzy logic, which has better performance in large-curvature segments and corners, but this method is still implemented in low-level controllers.

According to above discussions, this paper will firstly extend the contour error estimation method in [11], which does not require complicated computation and is easy to implement while ensuring high estimation accuracy. Then, the contour error compensation will be performed by using sILC to modify the reference trajectories. In order to improve the performance on large-curvature contour segments, we will develop a mechanism of feed rate adjustment, without changing the trajectory planning module. The combination of feed rate adjustment and sILC can achieve improved contour control, as will be illustrated by experimental results on a

This research is supported by National Key R&D Program of China under grant number 2019YFB1703700, National Natural Science Foundation of China under grant number U1913213 and UK Engineering and Physical Sciences Research Council under grant number EP/T006951/1.

J. Li, Y. Wang and Q. Shu are with the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, China.

D. Huang is with the School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China.

Y. Li is with the School of Engineering and Informatics, University of Sussex, Brighton, BN1 9RH, UK. Email: yl557@sussex.ac.uk

three-axis CNC machine tool.

## II. CONTOUR ERROR ESTIMATION AND COMPENSATION

The machining process and control framework in this paper are shown in Figure 1. The CNC machine tool includes decoding and trajectory planning module, data buffer, servo drive system, and watch data module. The algorithm in this paper includes contour error control module and feed rate adjustment module. In a typical machining process, G code is obtained according to the CAD (or CAM) model of the product. The CNC machine tool parses G code into a contour curve composed of small line segments, and then the machine tool's trajectory planning module uses them to form the reference trajectory of each axis. These reference trajectory points are stored in a buffer. The servo system obtains data from the buffer and sends instructions to each drive axis to generate motion. At the same time, the feedback position data is stored in the same buffer, and then exported by the watch data module as the input of the contour control module. The output of the contour control module is a geometric contour that does not include the feed rate. After the feed rate adjustment module, maximum feed rate information is added. Then, new G code can be obtained and sent to the machine tool for operation. This process is repeated until the contour accuracy meets the requirements.

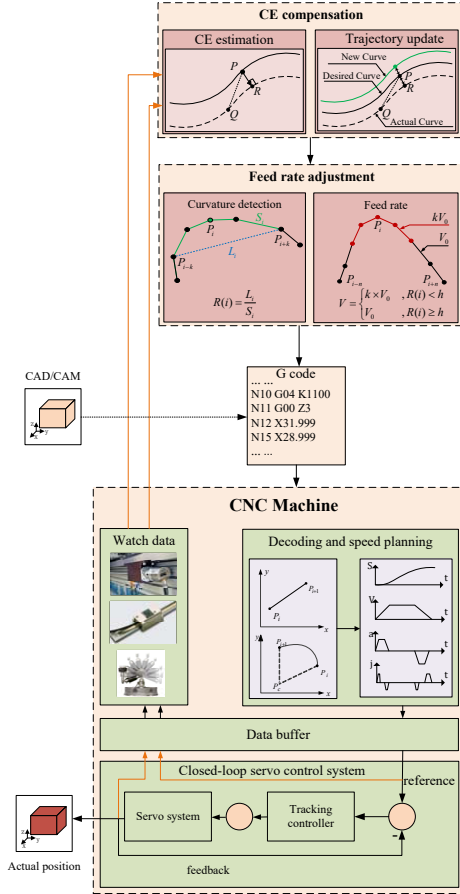


Fig. 1. CNC machining process and control framework

### A. Contour error estimation

In this section, the two-dimension contour error estimation method in [11] is firstly introduced and then expanded to estimate the contour error in a three-dimension case.

Denoting an incremental index  $s$ , the desired position of the machine tool can be described as

$$\vec{x}_d(s) = [x_{d1}(s), x_{d2}(s), \dots, x_{dn}(s)]^T \quad (1)$$

where  $n$  represents the number of axes involved in the motion of the drive system. Then, the actual position of the machine tool is described as

$$\vec{x}(s) = [x_1(s), x_2(s), \dots, x_n(s)]^T \quad (2)$$

Define  $\vec{e}$  as an error vector at the desired position corresponding to  $s_1$ , as below

$$\vec{e}(\hat{s}) = \vec{x}_d(s_1) - \vec{x}(\hat{s}) \quad (3)$$

where  $\hat{s}$  is the estimate of  $s_1$ . By searching  $\hat{s}$  to minimize the magnitude of  $\vec{e}(\hat{s})$ , we define this minimal magnitude as the contour error at the desired position corresponding to  $s_1$ .

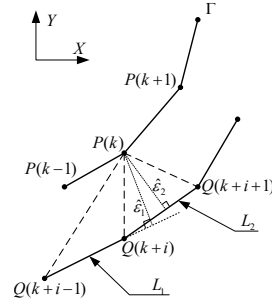


Fig. 2. Estimation of contour error for digital curves

In CNC systems, the desired and actual positions are usually in the form of digital curves (discrete data points). Therefore, based on the above estimation strategy and combined with the characteristics of digital curves, contour error estimation is illustrated in Figure 2. For any trajectory,  $Q(k)$  represents the actual tool position at time  $k$  while  $P(k)$  represents the desired position, where  $Q(k) = [x_{qk}, y_{qk}]^T$ ,  $P(k) = [x_{pk}, y_{pk}]^T$ .

First, we search the nearest actual position point from the desired one. When the traversal reaches the desired position point  $P(k)$ , we search the nearest actual position point  $Q(k+i)$  from  $P(k)$ . The distance between  $P(k)$  and  $Q(k)$  can be computed as

$$L = \sqrt{(x_{pm} - x_{qk})^2 + (y_{pm} - y_{qk})^2} \quad (4)$$

Then, by connecting  $Q(k+i-1)$ ,  $Q(k+i)$  and  $Q(k+i+1)$ , we get two segments  $L_1$  and  $L_2$ . Let  $L_a = \|\vec{Q(k+i)P(k)}\|$ ,  $L_b = \|\vec{Q(k+i)Q(k+i+1)}\|$ , and  $L_c = \|\vec{P(k)Q(k+i+1)}\|$ , which can be computed as shown in Figure 3. Then, we can compute  $\alpha_1$  and  $\alpha_2$ , and the coordinates of the perpendicular foot  $P_c(k)$ , as follows.

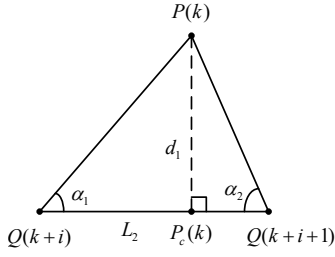


Fig. 3. Illustration of the perpendicular foot  $P_c(k)$

Case 1: If  $\alpha_1 < 90^\circ$  or  $\alpha_2 < 90^\circ$ , the perpendicular foot is on the segment and there are two equations about it:

$$\begin{cases} \|\overrightarrow{P(k)P_c(k)}\| = \|\overrightarrow{P(k)Q(k+i)}\| \sin \alpha_1 \\ \|\overrightarrow{Q(k+i)P_c(k)}\| = \|\overrightarrow{P(k)Q(k+i)}\| \cos \alpha_1 \end{cases} \quad (5)$$

so the coordinates of  $P_c(k)$  can be computed as

$$P_c(k) = Q(k+i) + \frac{\|\overrightarrow{Q(k+i)P_c(k)}\| \overrightarrow{Q(k+i)Q(k+i+1)}}{\|\overrightarrow{Q(k+i)Q(k+i+1)}\|} \quad (6)$$

Case 2: If  $\alpha_1 = 90^\circ$  or  $\alpha_2 = 90^\circ$ , there is no perpendicular foot on segment  $L_2$ .

Case 3: If  $\alpha_1 > 90^\circ$  or  $\alpha_2 > 90^\circ$ ,  $Q(k+i)$  can be regarded as the perpendicular foot.

Case 4: The same method is used to determine whether there is any perpendicular foot on segment  $L_1$ . If there are perpendicular feet on both  $L_1$  and  $L_2$ , we choose the point closer to  $P(k)$  as the perpendicular foot. If there is no perpendicular foot on either  $L_1$  or  $L_2$ , we choose  $Q(k+i)$  as the perpendicular foot, i.e.  $P_c(k) = Q(k+i)$ .

With  $P_c(k)$  as the perpendicular foot corresponding to the desired position point  $P(k)$ , the contour error in a 2-dimension case can be estimated as  $\varepsilon(k) = \|P(k) - P_c(k)\|$ .

The above contour error estimation method in a two-dimension case can be extended to a three-dimension (3D) case. We use  $P(i) = [x_i(t), y_i(t), z_i(t)]^T$  to denote a point on a 3D spline, where  $t$  is the spline parameter and  $A_i, B_i, C_i, D_i$  are the 3D coefficient vectors. Then, these coefficients can be determined for 3D Hermite spline curves, by solving

$$\begin{cases} x_i(t) = A_{i1}t^3 + B_{i1}t^2 + C_{i1}t + D_{i1} \\ y_i(t) = A_{i2}t^3 + B_{i2}t^2 + C_{i2}t + D_{i2}, \quad t \in [0, 1] \\ z_i(t) = A_{i3}t^3 + B_{i3}t^2 + C_{i3}t + D_{i3} \end{cases} \quad (7)$$

By calculating the distance between  $P(k)$  and the perpendicular foot  $P_m(k) = [x_{mk}, y_{mk}, z_{mk}]^T$  as

$$L = \sqrt{(x_{pm} - x_{qk})^2 + (y_{pm} - y_{qk})^2 + (z_{pm} - z_{qk})^2} \quad (8)$$

we can find the actual position point nearest to the desired one. Then the approximate perpendicular Point  $P_c(k)$  can be computed by Eq. (6).

As is shown in Figure 4, in order to find a more accurately approximated perpendicular foot on the spline curve, we

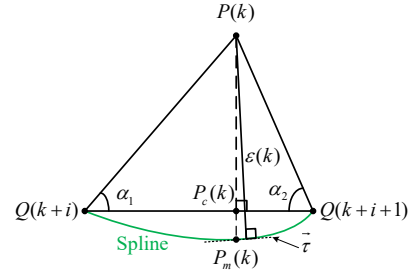


Fig. 4. Illustration of the perpendicular foot  $P_m(k)$  on a 3D spline curve

choose the spline parameter  $t$  as below:

$$t = \frac{\|\overrightarrow{Q(k+i)P_c(k)}\|}{\|\overrightarrow{Q(k+i)Q(k+i+1)}\|} \quad (9)$$

Then, with Eq. (7), the perpendicular foot  $P_m(k) = [x_k(t), y_k(t), z_k(t)]^T$  is calculated, and the tangent vector  $\vec{\tau}$  at point  $P_m(k)$  can be easily obtained according to the expression of the spline curve. Finally, the contour error is estimated as

$$\varepsilon(k) = \frac{\|\vec{\tau} \times \overrightarrow{P(k)P_m(k)}\|}{\|\vec{\tau}\|} \quad (10)$$

### B. Contour error compensation

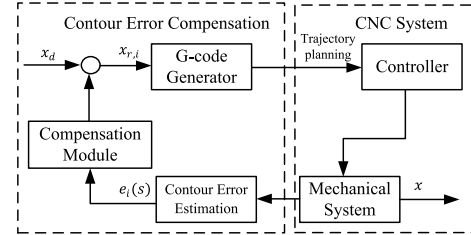


Fig. 5. Contour error compensation

Suppose the reference trajectory is  $\vec{x}_r(s)$ , then it can be modified on the basis of the estimation of the contour error, as below:

$$\vec{x}_{r,i}(s) = \vec{x}_{r,i-1}(s) + Q\vec{\varepsilon} \quad (11)$$

where  $x_{r,0}(s) = \vec{x}_d(s)$ ,  $\vec{x}_{r,i}(s)$  is the reference trajectory in the  $i$ -th iteration,  $Q$  is the learning rate and  $\vec{\varepsilon}$  is the contour error vector. G-code is then generated according to the reference trajectory, as shown in Figure 5.

### III. FEED RATE ADJUSTMENT BASED ON CURVATURE DETECTION

In CNC machining, the contour error is related to the feed rate and the curvature of the trajectory. Therefore, the curvature of the reference trajectory is detected to partition the trajectory into different segments, and the contouring accuracy of the trajectory is improved by feed rate adjustment.

### A. Curvature detection

In computer vision, there are many algorithms for corner detection, such as Rosenfeld and Johnston (RJ73), Freeman and Davis (FD77), Sarfraz, Asim and Masood (SAM06) in [24] and curve to chord ratio (CCR) in [25]. For digital curves, this paper adopts CCR for curvature detection as it is simple to implement, efficient and accurate. As shown in Figure 6, CCR firstly establishes the region of supports (RoSs) which are the two sides of the corner. Then based on the RoSs, the curvatures of these candidate corners are measured. The range of large-curvature segments in the trajectory can be detected only if appropriate RoSs domain and threshold value are set. RoSs domain is the neighborhood to be considered when the point is detected and the threshold value is set to determine whether the segment is a large-curvature segment. In general, the larger the range of RoSs domain is, the greater the influence of its neighborhood will be on the detection result of a certain point.

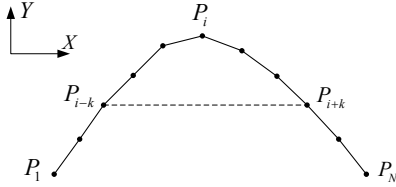


Fig. 6. Curvature detection by CCR

Suppose that the trajectory consists of a series of data points  $P_1(x_1, y_1), P_2(x_2, y_2), \dots, P_n(x_n, y_n)$ , then the chord ratio  $R(P_i)$  can be computed by

$$R(P_i) = \frac{\|P_{i-k}P_{i+k}\|}{\sum_{j=i-k}^{i+k-1} \|P_jP_{j+1}\|} \quad (12)$$

By setting an appropriate threshold and comparing the value of  $R(P_i)$  with the threshold, it is possible to detect whether the point is on a large-curvature segment.

### B. Feed rate adjustment

Assuming that the default feed rate is  $V_0$ , the feed rate is modified as follows:

$$V = \begin{cases} V_0, & \text{normal segments;} \\ KV_0, & \text{large-curvature segments.} \end{cases}$$

where  $K$  is a positive scalar, which is less than 1 for large-curvature segments.

The feed rate adjustment module based on curvature detection is integrated into the G-code generator in Figure 5. By reducing the feed rate, the contour error in the first iteration will be reduced for large-curvature segments, so as to improve the contour control accuracy in subsequent iterations.

## IV. EXPERIMENTS

In order to verify the effectiveness of the contour error compensation algorithm based on feed rate adjustment, experiments are conducted on an experimental platform shown in Figure 7. The CNC machine tool adopts a Googol CNC

system and an encoder with resolution of 32768 P/R. During the experiments, data collection is mainly completed by the control card with sampling frequency of 500Hz. The digital control system collects the encoder pulse signal, and then converts it to the position of the motor pitch for calculation. A computer is used to process the information to generate new G-code, and the CNC machine tool loads the new G-code, performs the task, and collects the actual tool position after one iteration. The data collected during the task are digital curves (discrete data points), and the generation of G-code is based on the method of G01 linear interpolation.

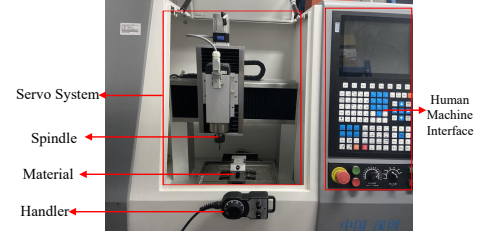


Fig. 7. Three-axis CNC machine tool

### A. Contour error compensation without feed rate adjustment

The experiments are firstly conducted with a fourth order ellipse trajectory shown in Figure 8. With the default feed rate of 1000mm/min and the iterative learning rate of 0.8, the results are obtained as shown in Figures 9 and 10. It is found that the contour error decreases significantly after the first iteration, and converges to a small range after 3 iterations. In particular, the maximum of the contour error decreases from about 14.5μm to 1.5μm.

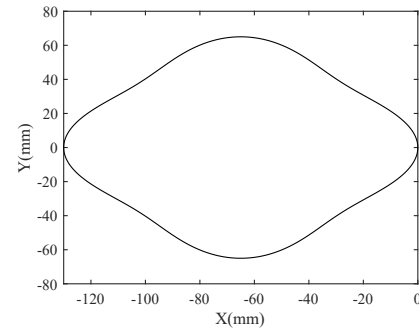


Fig. 8. Fourth order ellipse trajectory

For comparison, we consider a flower curve trajectory with large-curvature segments shown in Figure 11. The results in Figure 12 shows that the maximum of the contour error decreases from 26μm to about 7μm. As the number of iterations increases, the contour error decreases in most parts but not in those corresponding to large-curvature segments in Figure 11.

### B. Curvature detection

By setting the RoSs domain as 4 and the threshold value as 0.998, the curvature detection method in Section III.A is

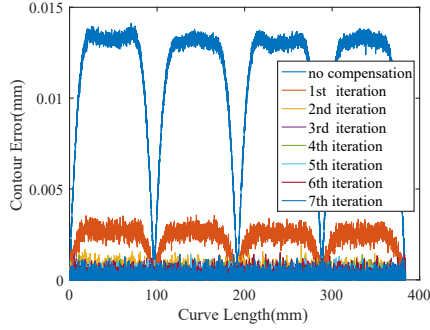


Fig. 9. Contour error compensation in different iterations

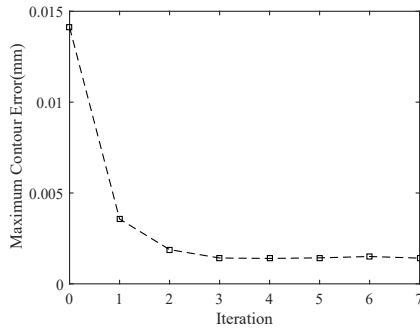


Fig. 10. Variation of maximum contour error

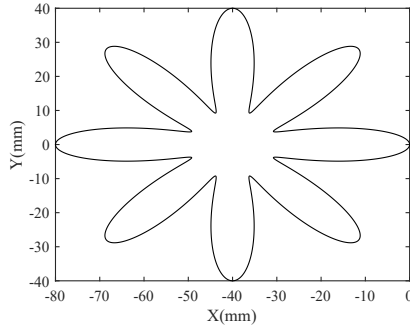


Fig. 11. Flower curve trajectory

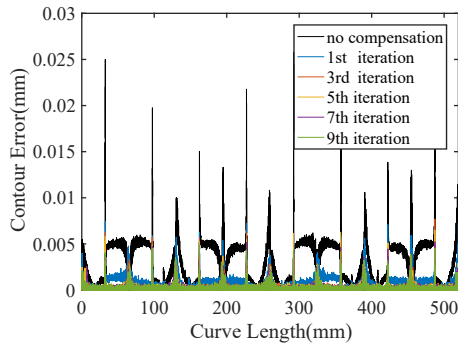


Fig. 12. Contour error compensation for the flower curve trajectory

used for the trajectories in Figures 8 and 11. From Figure 13(a), it is found that no large-curvature segment is detected in the fourth order ellipse trajectory. For the flower curve trajectory, there are 8 large-curvature segments detected as shown in Figure 13(b).

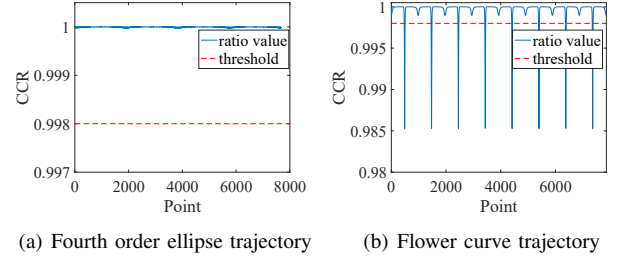


Fig. 13. Curvature detection results

### C. Contour control based on feed rate adjustment

In this section, we implement feed rate adjustment introduced in Section III.B. When the default feed rate is 1000mm/min, the adjustment rate  $K$  is set as 0.1, so the feed rate of a segment will be either 1000mm/min or 100mm/min (written as 1000-100mm/min). Since no large-curvature segments are found in the fourth order ellipse trajectory, feed rate adjustment is only applied to the flower curve trajectory, as shown in Figure 14. It can be found that the contour error at the large-curvature segments decreases significantly after the first iteration, with a maximum value of 9 $\mu$ m, much lower than that without feed rate adjustment. After 4 iterations, the maximum of the contour error decreases to 2.4 $\mu$ m, which is 7 $\mu$ m lower than that without feed rate adjustment. In conclusion, the compensation effect is no longer limited at large-curvature segments, with the proposed feed rate adjustment method.

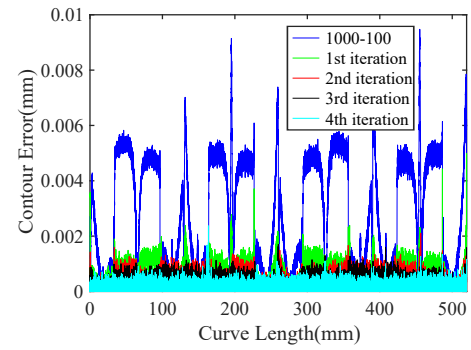


Fig. 14. Contour error compensation with the feed rate of 1000-100mm/min

### D. Machining efficiency

As reducing the feed rate affects the machining efficiency, we propose to increase the feed rate for smooth segments when decreasing the feed rate for detected large-curvature segments. By adjusting the feed rate to 1500-100mm/min, we obtain the experimental results in Figure 15. After increasing the default feed rate, the maximum contour error in the



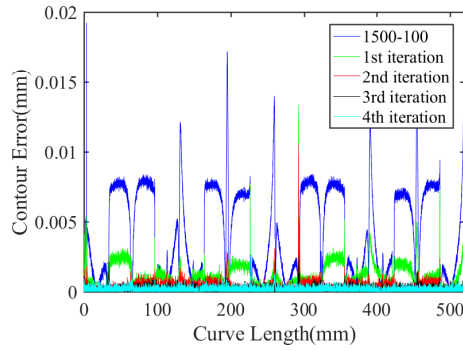


Fig. 15. Contour error compensation with the feed rate of 1500-100mm/min

first iteration increases to 19 $\mu$ m, but after 4 iterations, the maximum contour error still decreases to 2.4 $\mu$ m.

The machining time and the maximum contour errors for the flower curve trajectory are summarized in table I, which indicates the improvement of the contour control performance with maintained machining efficiency under the proposed method.

TABLE I  
CONTOUR ERROR COMPENSATION OF THE FLOWER CURVE

Feed rate (mm/min)	1000	1000-100	1500-100
$e_{max}^0(\mu m)$	25.784	9.213	19.312
$e_{max}(\mu m)$	7.240	2.357	2.468
machining time (s)	31.690	38.128	29.886

## V. CONCLUSION

In this paper, a contour error estimation method is extended and a compensation algorithm based on feed rate adjustment is proposed to address the problem of poor compensation performance for large-curvature trajectories. A curvature detection method is developed based on the curve to chord ratio (CCR). The experimental results show that the performance of the compensation algorithm at the large-curvature segments is significantly improved by adjusting the feed rate. The overall machining efficiency is maintained by increasing the feed rate at the smooth segments, so that high-speed and high-precision machining can be realized. Our future works include extending the proposed method to five-axis machine tools by considering rotations.

## REFERENCES

- [1] Z. Chen, B. Yao, and Q. Wang, " $\mu$ -synthesis-based adaptive robust control of linear motor driven stages with high-frequency dynamics: A case study," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1482–1490, 2015.
- [2] Z. Chen, Q. Wang, and B. Yao, "Adaptive robust precision motion control of linear motors with integrated compensation of nonlinearities and bearing flexible modes," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 965–973, 2013.
- [3] S.-S. Yeh and P.-L. Hsu, "Perfectly matched feedback control and its integrated design for multi-axis motion systems," *J. Dyn. Sys., Meas., Control*, vol. 126, no. 3, pp. 547–557, 2004.
- [4] C.-H. Yeung, Y. Altintas, and K. Erkorkmaz, "Virtual cnc system. part i. system architecture," *International Journal of Machine Tools and Manufacture*, vol. 46, no. 10, pp. 1107–1123, 2006.

- [5] L. Tang and R. G. Landers, "Multiaxis contour control—the state of the art," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 1997–2010, 2013.
- [6] G. T.-C. Chiu and M. Tomizuka, "Contouring control of machine tool feed drive systems: a task coordinate frame approach," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 1, pp. 130–139, 2001.
- [7] Y.-T. Shih, C.-S. Chen, and A.-C. Lee, "A novel cross-coupling control design for bi-axis motion," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 14, pp. 1539 – 1548, 2002.
- [8] J. Yang and Z. Li, "A novel contour error estimation for position loop-based cross-coupled control," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 643–655, 2011.
- [9] G. Zhao, H. An, and Q. Zhao, "Contour error coupled-control strategy based on line interpolation and curve interpolation," *JCP*, vol. 8, no. 6, pp. 1512–1519, 2013.
- [10] R. Shi, Y. Lou, Y. Shao, J. Li, and H. Chen, "A novel contouring error estimation for position-loop cross-coupled control of biaxial servo systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2197–2202.
- [11] J. Li, Y. Wang, Y. Li, and W. Luo, "Reference trajectory modification based on spatial iterative learning for contour control of two-axis nc systems," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1266–1275, 2020.
- [12] Y. Koren, "Cross-Coupled Biaxial Computer Control for Manufacturing Systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 4, pp. 265–272, 1980.
- [13] C. Hu, B. Yao, and Q. Wang, "Coordinated adaptive robust contouring controller design for an industrial biaxial precision gantry," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 728–735, 2010.
- [14] W. Chen, D. Wang, Q. Geng, and C. Xia, "Robust adaptive cross-coupling position control of biaxial motion system," *Science China Technological Sciences*, vol. 59, no. 4, pp. 680–688, 2016.
- [15] B. Zhang, R. Cao, and Z. Hou, "The model-free adaptive cross-coupled control for two-dimensional linear motor," *Transactions of the Institute of Measurement and Control*, vol. 42, no. 5, pp. 1059–1069, 2020.
- [16] R. Shi, X. Zhang, and Y. Lou, "Contouring error vector and cross-coupled control of multi-axis servo system," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2062–2067.
- [17] Z. Kuang, G. Sun, and H. Gao, "Simplified newton-based cee and discrete-time fractional-order sliding-mode cec," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 175–185, 2019.
- [18] K. L. Barton and A. G. Alleyne, "A cross-coupled iterative learning control design for precision motion control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1218–1231, 2008.
- [19] D. Hoelzle and K. Barton, "A new spatial iterative learning control approach for improved micro-additive manufacturing," in *2014 American Control Conference*, 2014, pp. 1805–1810.
- [20] D. J. Hoelzle and K. L. Barton, "On spatial iterative learning control via 2-d convolution: Stability analysis and computational efficiency," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1504–1512, 2016.
- [21] J. Liu, X. Dong, D. Huang, and M. Yu, "Composite energy function-based spatial iterative learning control in motion systems," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1834–1841, 2018.
- [22] J. Dong, T. Wang, B. Li, and Y. Ding, "Smooth feedrate planning for continuous short line tool path with contour error constraint," *International Journal of Machine Tools and Manufacture*, vol. 76, pp. 1 – 12, 2014.
- [23] P. Ngo and Y. Shin, "Milling contour error control using multilevel fuzzy controller," *The International Journal of Advanced Manufacturing Technology*, vol. 66, pp. 1641–1655, 2012.
- [24] M. Sarfraz, "Corner detection for curve segmentation," *Interactive Curve Modeling: With Applications to Computer Graphics, Vision and Image Processing*, pp. 209–240, 2008.
- [25] S. W. Teng, R. M. Najmus Sadat, and G. Lu, "Effective and efficient contour-based corner detectors," *Pattern Recognition*, vol. 48, no. 7, pp. 2185 – 2197, 2015.